

Downstream Side

Upstream Side

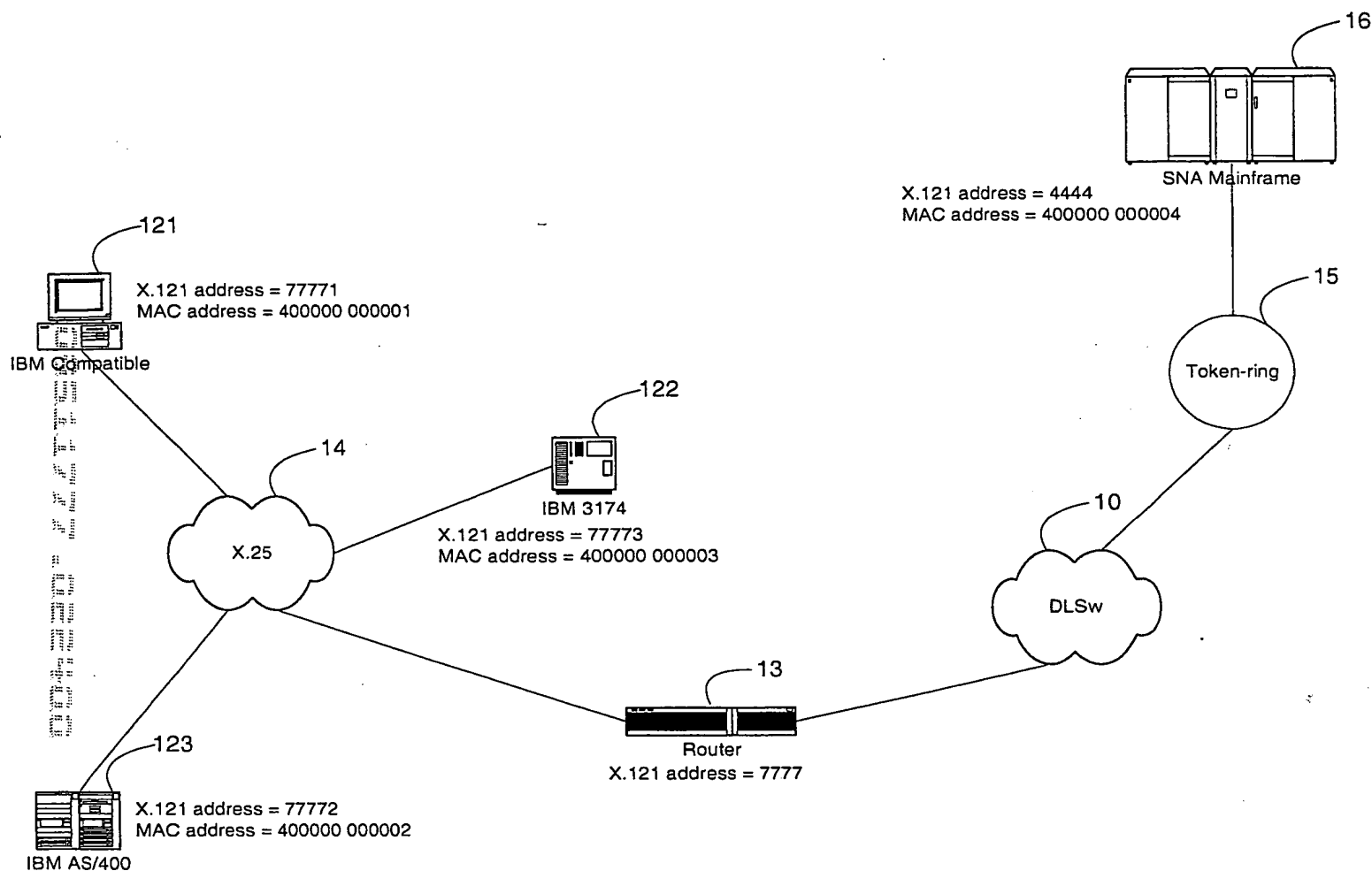


FIG. 1 100

PRIOR ART

Downstream Side

Upstream Side

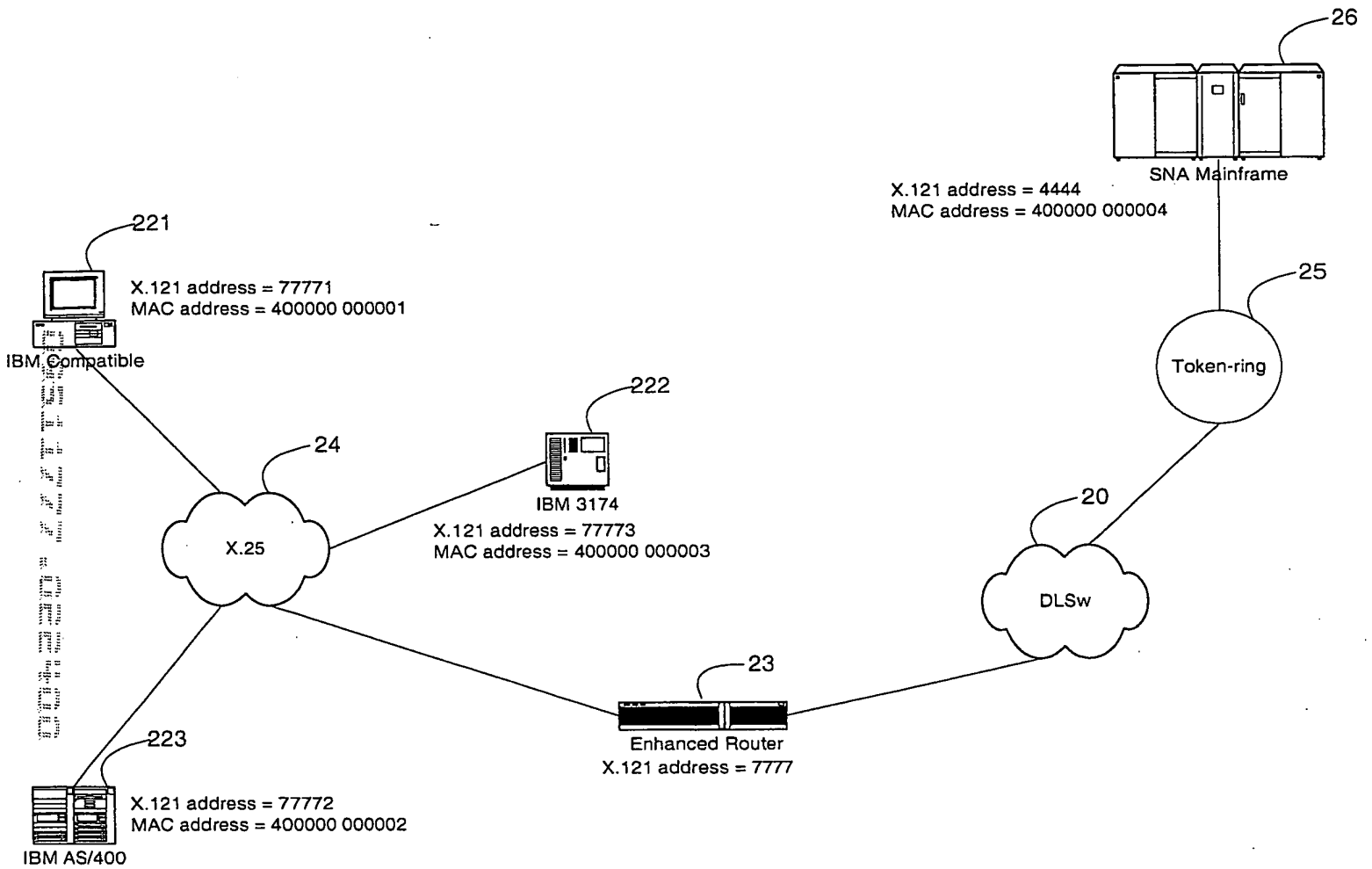


FIG. 2 200

Upstream Side

Downstream Side

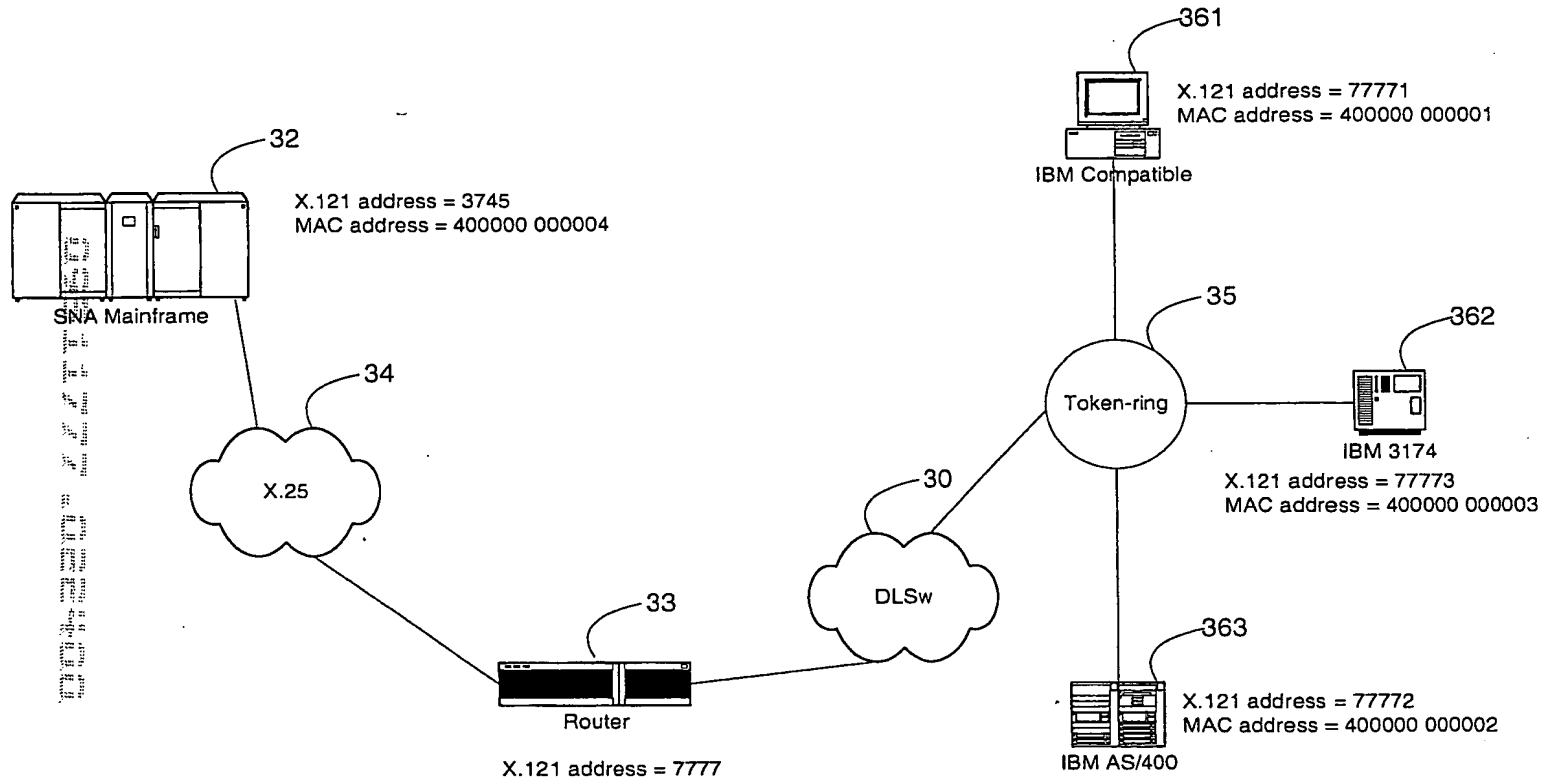


FIG. 3 300

PRIOR ART

Upstream Side

Downstream Side

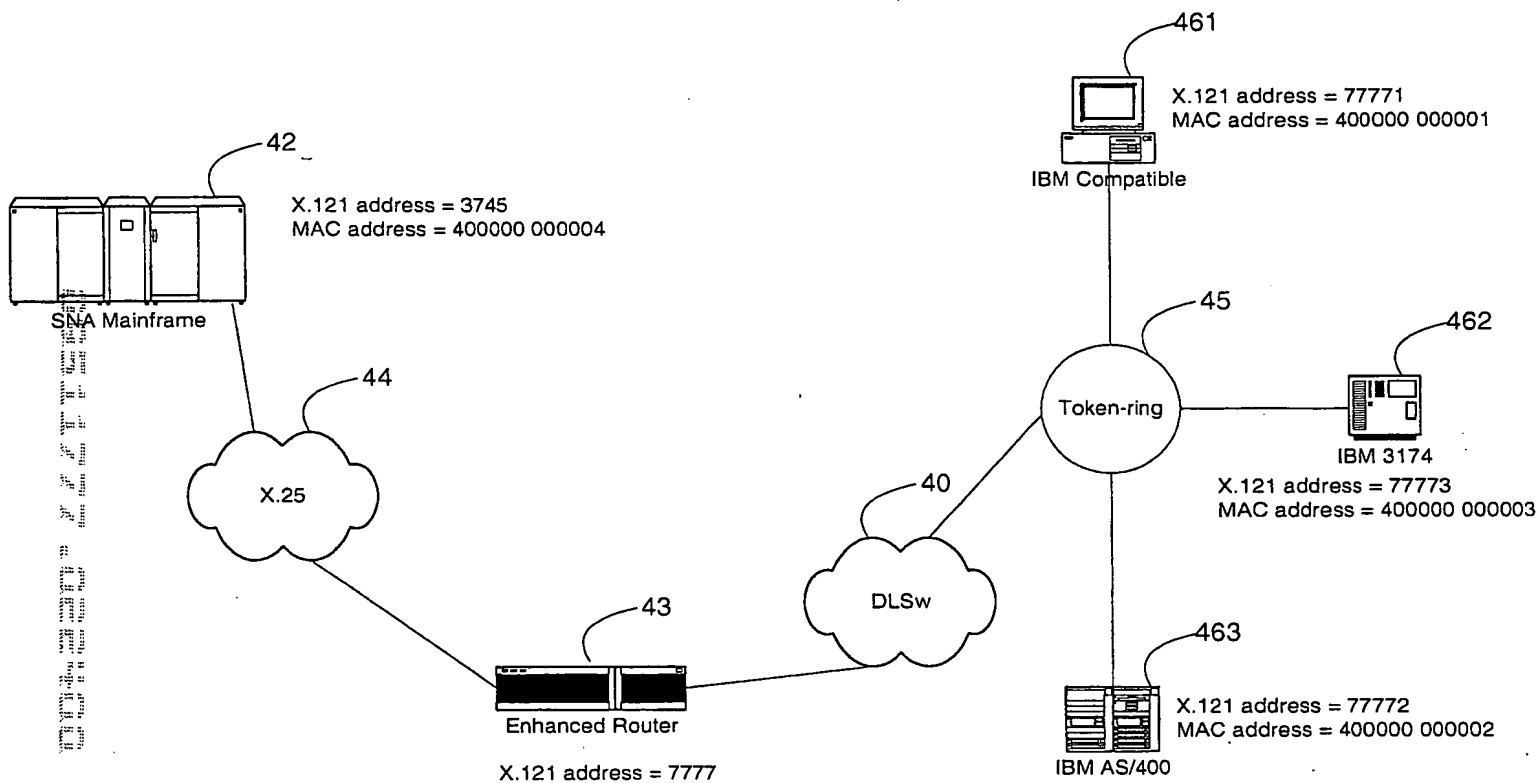


FIG 4 400

FIG. 5 is a block diagram of a network device 500, which may be a router or a switch, according to one embodiment. The network device 500 includes an X.25 interface 502, a switching/routing logic 504, a DLSw interface 506, an address pools 508, connection control logic 510, a dynamic routing table 512, an address configuration table 514, MIBs 516, and a management agent 518. The X.25 interface 502 is connected to an X.25 network. The DLSw interface 506 is connected to a DLSw network. The switching/routing logic 504 is connected to the X.25 interface 502 and the DLSw interface 506. The connection control logic 510 is connected to the X.25 interface 502 and the address pools 508. The dynamic routing table 512 is connected to the switching/routing logic 504 and the connection control logic 510. The address configuration table 514 is connected to the connection control logic 510 and the MIBs 516. The management agent 518 is connected to the MIBs 516.

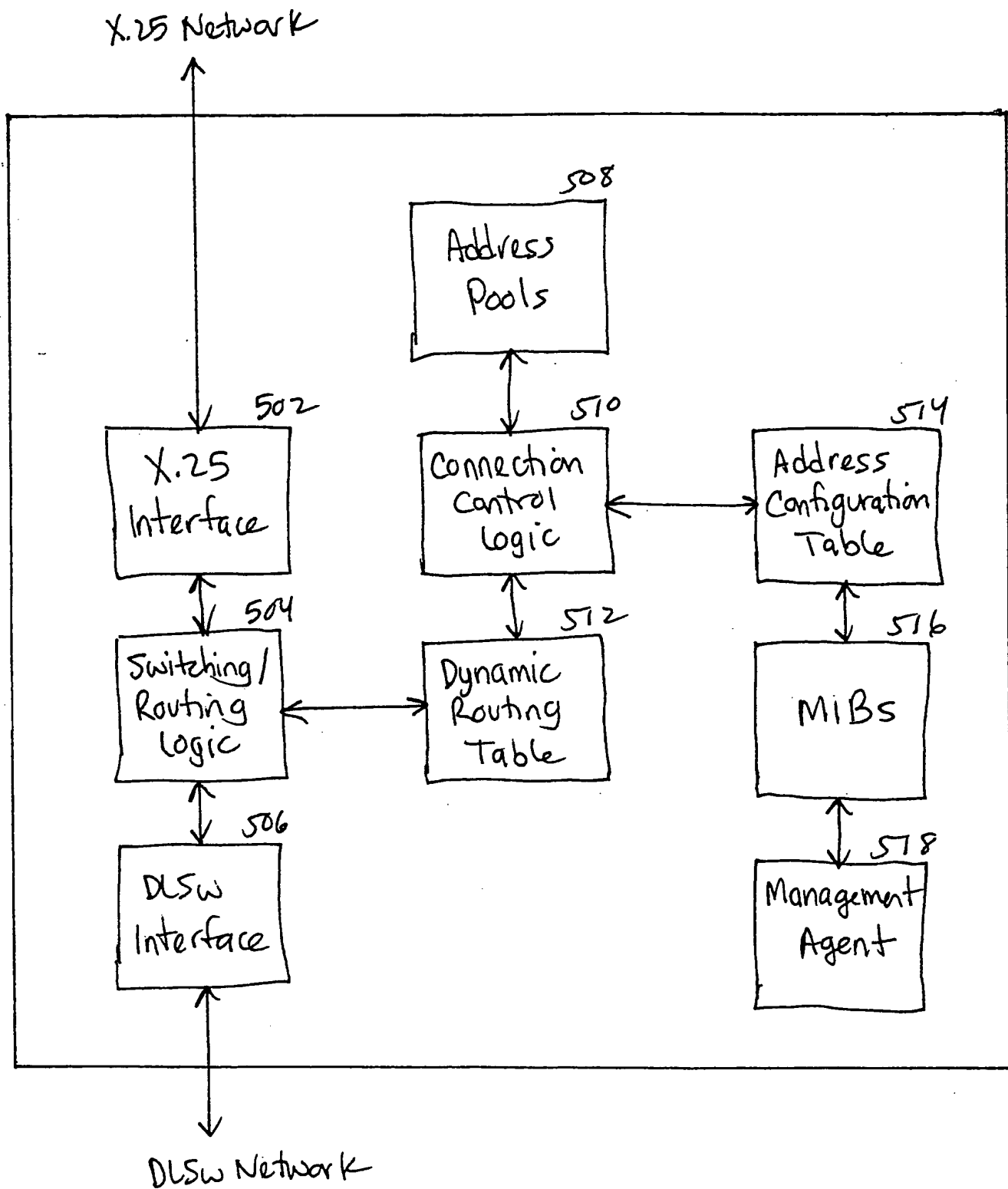


FIG. 5 500

Wildcard Character	Function
C	Matches any character, except those listed below. An ordinary character (like a, b, 7, or 5) matches only itself.
. (Dot or period) (. = hex 2E)	Matches a single character Example: PC._map Matches: PC1_map and PC2_map
^ (Caret or circumflex) (^ = hex 5E)	The caret or circumflex has special meaning only when it's the leftmost character in the wildcard expression. As such, it constrains the expression to match the leftmost portion of a line. A match of this type is called an "anchored match," because it is "anchored to a specific place in the line.
\$ (Currency symbol) (\$ = hex 24)	The currency symbol has special meaning only when it's the rightmost character in the wildcard expression. As such, it constrains the expression to match the rightmost portion of a line. A match of this type is called an "anchored match," because it is "anchored to a specific place in the line.
[c...] ([= hex 5B) (] = hex 5D)	As the rightmost element in a string, this element tells the wildcard to match any one of the characters enclosed in the brackets. Example: Map_[abc] Matches: Map_a, Map_b, or Map_c
[^c...] (^ = hex 5E)	When a caret is the first character of the bracketed string, the wildcard expression matches any character except those in the remainder of the string. Example: [^45678] Matches any character except 4, 5, 6, 7, or 8.
[l-r] (- = hex 2D)	The minus sign between two characters indicates a range of consecutive ASCII characters to match. This bracketed string of characters is known as a character class. Example: [0-9] Matches: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9

Rule/ Operator	Interpretation
? (Question Mark) (?=hex 3F)	A single-character regular expression followed by a question mark (?) matches zero or exactly one occurrence of that single-character regular expression. Example: [0-9]? Matches: any string of either zero or exactly one digit.
* (Asterisk) (*=hex 2A)	A single character regular expression followed by an asterisk (*) matches zero or more occurrences of that single-character regular expression. Example: [0-9]* Matches: any string of zero or more digits.
+ (Plus Sign) (+=hex 2B)	A single-character regular expression followed by a plus sign (+) matches one or more occurrences of that single-character regular expression. Example: [0-9]+ Matches: any string with one or more digits.
{m} {m,} {m,n} (Where m and n are integers) ({=hex 7B) (}=hex 7D) (,=hex 2C)	A one-character regular expression followed by {m}, {m,}, or {m,n} is a regular expression that matches a range of occurrences of the one-character regular expression. The values m and n must be non-negative integers less than 255. The symbols in braces mean the following: {m} matches exactly m occurrences {m,} matches at least m occurrences {m,n} matches any number of occurrences between m and n. Whenever a choice exists, the regular expression matches as many occurrences as possible.

Fig. 7

700

	802	804	806	808	810
	Map Name	Adjacent X.121 Address	Partner X.121 Address	Adjacent MAC Address	Partner MAC Address
812 -	AS400_map	77772	4444	400000000002	400000000004
814 -	3174_map	77773	4444	400000000003	400000000004
816 -	PC_map	77771	4444	400000000001	400000000004

FIG. 8 800
PRIOR ART

	902	904	906	908	910	911
	Map Name	Adjacent X.121 Address	Partner X.121 Address	Adjacent MAC Address	Partner MAC Address	Flag
912 -	Host1_map	.*	4444	(nil)	400000000004	0

FIG. 9 900

	1004	1006	1007	1008	1010
	Adjacent X.121 Address	Partner X.121 Address	X.25 LCN	Adjacent MAC Address	Partner MAC Address
1012 -	77772	4444	1	400000000005	400000000004

FIG. 10 1000

	1102	1104	1106	1108	1110
	Map Name	Adjacent X.121 Address	Partner X.121 Address	Adjacent MAC Address	Partner MAC Address
1112 -	AS400_map	3745	77772	400000000004	4000000000002
1114 -	3174_map	3745	77773	400000000004	4000000000003
1116 -	PC_map	3745	77771	400000000004	4000000000001

FIG. 11 1100
PRIOR ART

	1202	1204	1206	1208	1210	1211
	Map Name	Adjacent X.121 Address	Partner X.121 Address	Adjacent MAC Address	Partner MAC Address	Flag
1212 -	NPSI_host	3745	7777	400000000004	*	1

FIG. 12 1200

	1304	1306	1307	1308	1310
	Adjacent X.121 Address	Partner X.121 Address	X.25 LCN	Adjacent MAC Address	Partner MAC Address
1312 -	3745	7777	2	400000000004	4000000000002

FIG. 13 1300

	1402	1404	1406	1408	1410	1411
	Map Name	Adjacent X.121 Address	Partner X.121 Address	Adjacent MAC Address	Partner MAC Address	Flag
1412 -	NPSI_host	3745	(nil)	400000000004	*	1

FIG. 14 1400

	1504	1506	1507	1508	1510
	Adjacent X.121 Address	Partner X.121 Address	X.25 LCN	Adjacent MAC Address	Partner MAC Address
1512 -	3745	8888	1	400000000004	4000000000002

FIG. 15 1500

operator	Interpretation
(...) (Regular expression(s) enclosed in parentheses)	A regular expression enclosed within parentheses matches whatever the unadorned regular expression matches. You use parentheses to group a series of regular expressions that you want to have treated as a single-character regular expression. Example: 0[Xx]? Matches: 0, 0X, or 0x Example: (0[Xx])? Matches: nothing, 0X, or 0x.
((=hex 28) ()=hex 29)	You can have up to nine such substrings in a regular expression, and you can nest parentheses.
(Vertical Bar) (= hex 7c)	Two regular expressions separated by the vertical bar () match either a match for the first or a match for the second. Example: 7777(12 16) Matches: 777712 or 777716
[] (Square Brackets) ([=hex 5B) (]=hex 5D)	Matches any single character in the bracketed set. Example: 40000000000[123] Matches: 400000000001, 400000000002, or 400000000003

Fig. 16

1600